第 43 卷　第 2 期
2023 年 4 月

桂林电子科技大学学报
Journal of Guilin University of Electronic Technology

Vol. 43，No. 2
Apr. 2023

一种模糊互模拟的局部算法

胡晋玮，钱俊彦

（桂林电子科技大学 计算机与信息安全学院，广西 桂林　541004）

摘　要：为了快速地对模糊迁移系统中给定状态是否满足互模拟关系进行验证，提出了一种模糊互模拟的局部算法。该算法将验证与遍历相结合，在对状态是否满足模糊互模拟关系验证的同时，动态地增加状态空间，使得算法只需遍历部分状态空间即可完成验证。在部分情况下，尤其是当 2 个状态不满足模糊互模拟关系时，模糊互模拟的局部算法可以更快地对给定状态是否满足模糊互模拟关系进行验证。通过 Java 实现了模糊互模拟的局部算法和已有全局算法，并进行了比较实验。实验结果表明，在给定状态对不满足互模拟关系的情况下，本算法比现有模糊互模拟的全局算法的效率更高。

关键词：互模拟；模糊集；模糊迁移系统；模糊互模拟；局部算法

中图分类号：TP301　　**文献标志码**：A　　**文章编号**：1673-808X(2023)02-0102-06

A local algorithm of fuzzy bisimulation

HU Jinwei，QIAN Junyan

（School of Computer and Information Security，Guilin University of Electronic Technology，Guilin 541004，China）

Abstract：In order to verify whether the given states satisfiy the bisimulation，a local algorithm of fuzzy bisimulation is proposed. The algorithm takes verification and traversal at the same. While verify the given states whether satisfiy the fuzzy bisimulation，the state space is dynamically increased，so that the algorithm only needs to traverse part of the state space to complete the verification. In some cases，the local algorithm of fuzzy bisimulation can verify whether the given states satisfiy the fuzzy bisimulation more quickly，especially when the two states do not satisfy the fuzzy bisimulation. The local algorithm and the existing global algorithm are implemented by Java and compared by experiments. The experiments，shows that this algorithm is more efficient than the existing global fuzzy bisimulation algorithm when the given states do not satisfiy the bisimulation.

Key words：bisimulation；fuzzy set；fuzzy transition systems；fuzzy bisimulation；local algorithm

互模拟作为描述系统之间行为等价的重要概念，是解决状态空间爆炸问题的有效手段[1]。1987 年，Paige 等[2]在划分精化的思想下，给出了时间复杂度为 $O(m\log n)$ 的互模拟等价类划分算法，该算法需要对整个迁移系统的状态空间进行预先存储，属于互模拟等价验证全局算法的一种。然而，在很多情况下，并不需要求得整个迁移系统状态空间的互模拟等价类，只需对迁移系统中的给定状态是否满足互模拟关系进行判断。

1989 年，Fernandez 等[3]提出了互模拟等价关系判断的局部算法，该算法又称为“on the fly”算法。“on the fly”算法通过构建 2 个迁移系统的同步积，并对要判定状态对的后续状态对进行深度优先或广度优先搜索遍历，同时判断每对状态对是否满足互模拟关系，该算法的时间复杂度为 $O(n^2)$，其中 n 为 LTS 的状态数。Du 等[4]在“on the fly”算法的基础上提出了介于局部算法与全局算法之间的互模拟等价验证的准局部算法，通过对状态添加状态 0 或 1 的方法，避免了对状态的重复访问，降低了算法的时间复杂度。Lin[5]将局部算法应用于值传递系统中，Deng 等[6]将局部算法应用于概率迁移系统，并提出了概率迁移系统的局部算法。

收稿日期：2021-03-07

基金项目：国家自然科学基金(61562015)；广西自然科学基金(2018GXNSFDA138003)

通信作者：钱俊彦(1973—)，男，教授，博士，研究方向为软件工程、程序验证与测试、容错技术。E-mail:qjy2000@gmail.com

引文格式：胡晋玮，钱俊彦. 一种模糊互模拟的局部算法[J]. 桂林电子科技大学学报，2023，43(2)：102-107.

模糊集合是经典集合的扩展，主要用于对事物的不确定性进行描述，可有效地对主观概念和不确定性概念进行分析[7]。将模糊论的定量分析应用到系统分析中，可很好地分析系统能够在多大程度上满足系统的性质规约，对模糊自动机[8-10]、模糊 Petri 网[11-12]及模糊迁移系统[13-16]的研究已经取得了很多学术成果。

值得注意的是，当用模糊自动机构建系统模型时，所构造出来的模型并非唯一。为了更好地对这些模型是否等价进行判断，计算机科学工作者将互模拟的定义引入模糊迁移系统。Petković 等[17]通过代数的方法提出了有限模糊自动机全等概念。Bucholz[18]和 Sun 等[19]对有限模糊自动机和有限模糊迁移系统的互模拟进行了研究。Cao 等[20]针对无限迁移系统，提出了关于一般模糊迁移系统的模糊互模拟的定义。为了更好地对模糊迁移系统行为相似程度进行刻画，Cao 等[21]提出了行为距离的概念，当 2 个模糊迁移系统完全满足模糊互模拟性质时，其行为距离为 0。Wu 等[22]在算法和逻辑 2 个角度上对模糊互模拟进行了刻画。

为了更好地对模糊迁移系统是否满足模糊互模拟关系进行判断，Ćirić 等[23]给出了有限模糊自动机的模糊互模拟等价验证算法；Ignjatović[23]提出了对 2 个模糊社交网络进行模糊互模拟等价验证的算法，该算法的时间复杂度为 $O(lc^5)$，其中 l 为在计算过程中出现的不同模糊值的数量，c 为社交网络中节点数；Wu 等[22]提出了一种模糊互模拟等价验证的全局算法，该算法的时间复杂度为 $O(m^2n^4)$，其中 n 为模糊迁移系统中状态的个数，m 为模糊迁移系统中迁移的数量。

尽管互模拟在模糊领域已经取得了一定成果，但到目前为止，对模糊互模拟局部算法的研究尚未得到广泛关注。为了更快地对指定状态是否满足模糊互模拟关系进行验证，基于文献[5-6]的算法思想，提出了一种模糊互模拟等价关系验证的局部算法，以提高模糊互模拟等价验证的效率，并通过实验的方式与现有的模糊互模拟检验算法进行了比较。

1　预备知识

设 S 为非空集合，S 上的模糊集合 P 为 S 到 $[0,1]$ 的一个映射，$\mu:S\to[0,1]$，用 $F(S)$ 表示集合 S 上的全体模糊集，$\mu\in F(S)$，则模糊集 μ 的支撑集 $\mathrm{supp}(\mu)$ 定义为

$$\mathrm{supp}(\mu)=\{s\in S:\mu(s)>0\}。$$

若 $\mathrm{supp}(\mu)$ 是有限的，则模糊集 μ 可用 Zedeh 形式表示为

$$\mu=\frac{\mu(s_1)}{s_1}+\frac{\mu(s_2)}{s_2}+\cdots+\frac{\mu(s_n)}{s_n}。$$

对于任意的 $\mu\in F(S)$ 及 $U\subseteq S$，$\mu(U)=\sup\limits_{s\in U}\mu(s)$。特别地，$\mu(\varnothing)=0$。

设 μ,ν 为集合 S 上的 2 个模糊集合，且 $\mu\subseteq\nu$，对任意的 $s\in S$，均有 $\mu(s)\leqslant\nu(s)$。

定义 1[9]　模糊迁移系统（fuzzy transition system，简称 FTS）是一个三元组 $(S,A,\rightarrow)$，其中：

1) S 为状态的集合；

2) A 为动作的集合；

3) $\rightarrow\subseteq S\times A\times F(S)$ 为迁移关系。

定义 2[22]　给定集合 S，$R\subseteq S\times S$，若存在关于 R 的权重函数 $e:s\times s\to[0,1]$ 满足如下条件：

1)对于任意的 $s\in S$，$\mu(s)=\sup\limits_{s\in S}e(s,t)$；

2)对于任意的 $t\in S$，$\nu(t)=\sup\limits_{t\in S}e(s,t)$；

3)若 $e(s,t)>0$，则有 $(s,t)\in R$。

则称 $R^{\dagger}\subseteq F(S)\times F(S)$ 为关系 R 的提升，使得 $\mu R^{\dagger}\nu$ 成立。

定理 1[22]　令 S 为一个集合，$R\subseteq S\times S$，$\mu,\nu\in F(S)$。若 $\mu R^{\dagger}\nu$ 成立，则对任意的 $s\in S$，均有

1) $\mu(s)\leqslant\nu(\overrightarrow{R_s})$，

2) $\nu(s)\leqslant\mu(\overleftarrow{R_s})$，

其中，$\overrightarrow{R_s}=\{s':sRs'\}$，$\overleftarrow{R_s}=\{s':s'Rs\}$。

定义 3[22]　若对于任意的 $R\in S\times S$，sRt 均满足以下条件：

1)对于任意的 $s\times\alpha\rightarrow\mu$，均有 $t\times\alpha\rightarrow v$，使得 $\mu R^{\dagger}\nu$ 成立；

2)对于任意的 $t\times\alpha\rightarrow v$，均有 $s\times\alpha\rightarrow\mu$，使得 $\nu R^{\dagger}\mu$ 成立。

则称 R 为互模拟关系。

若存在一个模糊互模拟关系 R，使得 $(s,t)\in R$，则称状态 s,t 满足模糊互模拟等价关系，记为 $s\sim t$。

2　模糊互模拟验证的局部算法

在递归定义的基础上，结合“On The Fly”算法和对提升关系 $R^{\dagger}$ 的判断算法，给出模糊迁移系统的互模拟等价验证的局部算法。

在模糊迁移系统中，互模拟可采用递归的方式进行定义。

定义 4[22]　设 FTS 为一个三元组 $M=(S,A,\rightarrow)$，定义如下：

1) $R_0=S\times S$。

2)对于任意的 $n>0$，$sR_{n+1}t$，若

a)对于任意的 $s\times\alpha\rightarrow\mu$，存在 $t\times\alpha\rightarrow v$，使得 $\mu R_n^{\dagger}\nu$ 成立；

b)对于任意的 $t\times\alpha\rightarrow v$，存在 $s\times\alpha\rightarrow\mu$，使得 $\mu R_n^{\dagger}\nu$ 成立。

3) $\sim=\bigcap_{n\geqslant 0}R_n$。

命题 1 在有限 FTS 中，定义 3 与定义 4 所定义的模糊互模拟是相同的。

证明 为了方便表示，用 $\sim_{\mathrm{def4}}$ 表示定义 4 中所定义的模糊互模拟，用 $\sim_{\mathrm{def3}}$ 表示定义 3 中表示的模糊互模拟，则

1)根据 $\sim_{\mathrm{def3}}$ 的定义可知，若 $s\sim_{\mathrm{def3}}t$，则有 $s\sim_{\mathrm{def4}}t$ 成立。

2)设 $s\sim_{\mathrm{def4}}t$，且 $s\times\alpha\rightarrow\mu$，要证 $\sim_{\mathrm{def4}}$ 为模糊互模拟关系，即要证存在 $t\times\alpha\rightarrow v$，使得 $\mu\sim_{\mathrm{def4}}^{\dagger}\nu$ 成立。设集合 $D=\{\nu:t\times\alpha\rightarrow v\wedge\mu\not\sim_{\mathrm{def4}}^{\dagger}\nu\}$，即对于任意的 $\nu\in T$，均有 n_ν 使得 $\mu\not\sim_{n_\nu}^{\dagger}\nu$。设 $n_{\max}=\max\{n_\nu:\nu\in T\}$，则有 $\mu\not\sim_{n_{\max}}^{\dagger}\nu$。又因为 FTS 是有限的，故集合 D 是有限的。因为 $s\sim_{\mathrm{def4}}t$ 成立，则有 $s\sim_{n_{\max}+1}t$，即存在 ν，$t\times\alpha\rightarrow v$，使得 $\mu\sim_{n_{\max}}^{\dagger}\nu$ 且 $\nu\notin T$。故对于任意的 $s\times\alpha\rightarrow\mu$，均存在 ν 使得 $\mu\sim_{\mathrm{def4}}^{\dagger}\nu$ 成立。同理，可证明若 $t\times\alpha\rightarrow v$，则存在 $s\times\alpha\rightarrow\mu$，使得 $\mu\sim_{\mathrm{def4}}^{\dagger}\nu$ 成立。

文献[22]提出了对于给定的集合 $S,\mu,\nu\in F(S)$，$R\subseteq S\times S$，判断 $\mu R^{\dagger}\nu$ 的算法。该算法时间复杂度为 $O(n^2)$。在此基础上，结合“On the Fly”算法的思想，提出了关于模糊互模拟验证的局部算法。

在模糊互模拟局部算法中，用集合 Errors 存储在算法执行过程中被判定为不具有互模拟性质的状态对；用数组 Visited 表示已经被遍历过的状态对。若一对状态已经被多次遍历，但其是否满足模糊互模拟关系尚不能确定时，则先假设其满足互模拟关系，同时用数组 LetBis 对其进行存储，若在后续的遍历过程中发现该状态对不满足互模拟关系，则将该状态对从数组 LetBis 中删除，同时将其添加到数组 Errors 中，并重新执行函数 checkBis。为了保证状态遍历和判断的先后顺序，用栈 Stack 存储尚未完成模糊互模拟关系判断的状态对。

算法 1 模糊互模拟检测的局部算法

输入：模糊迁移系统 FTS 以及状态对 (s,t)

输出：s,t 是否满足模糊互模拟关系

```
Errors=∅
isBis=unknow
while isBis=unknow
    isBis=checkBis(s,t)
return isBis
Function checkBis(s,t)
Visited=∅
LetBis=∅
Stack=∅
Push (s,t) into Stack
While Stack≠∅
    (s,t)top(Stack)
    Visited=Visited∪(s,t)
    if act(s)==act(t) then
      outer
      for all α∈act(s)
        for all s×α→μ
          for all t×α→ν_t
            R=∅
            for all s_i∈ supp(μ_i)
              for all t_j∈ supp(ν_j)
                if (s,t) Errors then
                  R=R
                else if (s,t)∈ Visited then
                  R=R∪(s,t)
                  LetBis LetBis∪(s,t)
                else
                  Push (s_i,t_j) into Stack
                  Break outer
            b_{i,j} check (μ_i,ν_j,R)
          actionBis=(∧_i(∨_j b_ij))∧(∧_j(∨_i bij))
          isBis=∧_{α∈A} B
      else
          isBis=false
      if isBis=false then
        Errors=Errors∪(s,t)
          if (s,t) LetBis then
            return unknow
      if isBis=true then
      return true
else
return false
```

在算法 1 的执行过程中，将集合 Errors 设置为全局变量，并将其设置为空集，将集合 Visited、LetBis 和 Stack 设置为函数 checkBis (s,t) 的局部变量，且每次执行函数 checkBis (s,t) 时，Visited、LetBis 和 Stack 都将被置为空集。

设 (s_0,t_0) 为要验证是否满足模糊互模拟关系

的状态对，(s,t) 为 (s_0,t_0) 的某个后继状态对。在函数 checkBis (s_0,t_0) 中，当 (s,t) 第 1 次被访问时，将其加入集合 Visited 中。若 (s,t) 具有不同的动作，则其不满足模糊互模拟关系，将其加入集合 Errors 中；若 (s,t) 具有相同的动作，则对其后继状态继续进行遍历。当 (s,t) 已经存在于集合 Visited 中时，将其加入集合 LetBis 中，并在对 (s_0,t_0) 进行模糊互模拟关系判断时添加到关系集合 R 中。若在后续的遍历过程中，发现状态 $(s,t)\in$ LetBis 不满足模糊互模拟关系，则将其从集合 LetBis 中删除，并加入集合 Errors 中，同时返回结果 unknow，重新执行函数 checkBis (s_0,t_0)。

定理 2　设 FTS1、FTS2 为 2 个有限模糊迁移系统，s、t 分别为这 2 个迁移系统中的状态，则只有 $s\sim t$ 时，算法 1 的返回结果为 true。

证明　1)算法 1 的可终止性证明。令 Bis_i 为第 i 次执行函数 checkBis_i，Errors_i、LetBis_i 分别为执行完 checkBis 之后的 Errors、LetBis 集合。由算法 1 的第 9～12 行和第 31～36 行可知，当 Bis_i 执行结束时，会产生 2 种可能，即假设错误，执行算法 1 第 34～37 行或直接跳过执行第 38 行语句。对于前一种可能，由算法 1 的第 32、33 行可知，$\text{Errors}_i\cap\text{LetBis}_i\neq\varnothing$；而后一种可能，算法则执行完毕。由算法 1 的第 21～27 行可知，$\text{Errors}_{i-1}\cap\text{LetBis}_i=\varnothing$。根据算法 1 的第 27 行可知，$\text{Errors}_{j-1}\subseteq\text{Errors}_i$。假设在第 j 次执行完 Bis 后，s、t 所有可达的非互模拟状态都已遍历完，则必有 $\text{Errors}_j=\text{Errors}_{j+1}$，再由算法 1 的第 32、33 行可知，算法 1 一定会终止。

2)算法 1 的正确性证明。设 Visited_i 为算法 1 第 i 次执行函数 Bis 所产生的 Visited 集合，$R_i=\text{Visited}_i-\text{Errors}_i$。同时，假设算法在执行 k 次函数 checkBis 后终止。由定义 4 可知，对于任意的 $i<k$，随着 i 不断增大，R_i 越来越接近模糊互模拟关系 $\sim$。在该情况下，对于任意的 $(s_i,t_j)\in R_i$，s_i 与 t_j 不确定是否满足互模拟状态，仅假设为互模拟状态，需要通过不断地遍历才可以确定是否满足模糊互模拟关系，但对于任意的 $(s_i,t_j)\notin R_i$，必有 s_i 与 t_j 不满足模糊互模拟关系。因此，当 checkBis_k 的返回值为 false 时，s 与 t 不满足模糊互模拟关系。当 Bis_k 的返回值为 true 时，必有 checkBis 为 true，即对于任意的 $\alpha\in A$，$s\times\alpha\rightarrow\mu$，$t\times\alpha\rightarrow v$，均有 $\mu R^{\dagger}\nu$。由定义 3 可知，状态对 (s,t) 满足模糊互模拟关系。

3　数据结构和算法时间复杂度分析

对于任意的模糊迁移系统，采用邻接链表的方式对其进行存储，其中模糊分布采用 Map 类型的链表进行存储。存储模糊迁移系统的数据结构如图 1 所示。

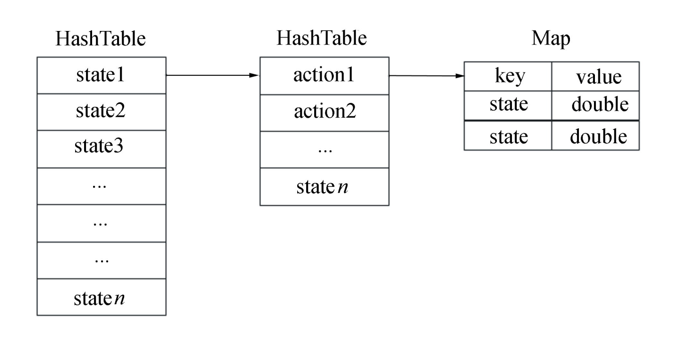


图 1　存储模糊迁移系统的数据结构

在图 1 所示的存储结构中，使用 HashTable 对状态和迁移键值对以及动作和模糊分布键值对进行存储，使用 Map 类型的 List 对模糊分布进行存储，其中 Map 的 key 值的数据类型与 HashTable 中状态的数据类型保持一致，value 类型则设置为 double 类型。为了方便状态对的存储，设置 StatePair 类对状态对进行存储，StatePair 类中主要包含属性状态 s、t 以及构造方法 StatePair (s,t)，s、t 的类型与最外层 HashTable 中的 key 值类型相同。集合 Visited、LetBis 和 Errors 均为 StatePair 类型的链表。

定理 3　模糊互模拟局部算法的时间复杂为 $O(m_1m_2n_1^3n_2^3)$，空间复杂度为 $O(n_1n_2)$。

证明　由算法 1 的第 1～5 行可知，该算法的时间复杂度主要由函数 checkBis 决定。设 s 和 t 为 checkBis 中的 2 个状态，模糊迁移系统 FTS1 的状态个数为 n_1，迁移数为 m_1，模糊迁移系统 FTS2 的状态数为 n_2，迁移数为 m_1。

1)当状态 s、t 为 FTS1、FTS2 的初始状态时，需要对 FTS1、FTS2 的全部状态进行遍历，此时共可构建 n_1n_2 个状态对。在函数中，栈 Stack 用于存储尚未判断是否满足模糊互模拟关系的状态对。根据函数的第 5 行可知，在最坏情况下，第 1 次执行函数 checkBis 时，所有遍历的状态对都进入栈中，该 while 循环至多执行 n_1n_2 次。根据函数的第 31～36 行可知，在最坏情况下，每执行 1 次 checkBis 函数，最多可以找到 1 对不满足模糊互模拟的状态。因此，在最坏情况下，第 2 次执行 checkBis 函数时，函数第 5 行的 while 循环需要执行 n_1n_2-1 次，第 3 次执行 checkBis 函数时，while 循环需要执行 n_1n_2-2 次。故有

$$\sum_{i=1}^{n_1n_2}n_1n_2-i=(n_1n_2)^2+\frac{(n_1n_2)(n_1n_2-1)}{2}\leqslant$$

$$\frac{3(n_1n_2)^2}{2}。$$

根据函数的第 15～31 行可知，对一个状态对进行检测时，最多对 m_1m_2 个迁移进行遍历。又因为 check (μ_i,ν_j,R) 的时间复杂度为 $O(n_1n_2)$，根据时间复杂度计算乘法原则，可得该算法的时间复杂度为 $O(m_1m_2n_1^3n_2^3)$。

2)根据 checkBis 的第 22～25 行及 34～35 行可知，当执行完一次 checkBis 函数时，集合 Visited 中的元素为集合 LetBis 与集合 Errors 中元素的总和，即 Visited＝LetBis ∪ Errors。在最坏情形下，执行 1 次 checkBis 时，所有的状态均被遍历，即 Visited 中包含 n_1n_2 个状态对时，算法的空间复杂度最高，故该算法的空间复杂度为 $O(n_1n_2)$。

4 算法的比较与实现

在对相关的理论进行分析后，通过实验比较局部算法与全局算法[22]的实际效率。

令 FTS 为一个模糊迁移系统，s、t 为迁移系统中的任意 2 个状态。设 FTS 中的状态数为 n，迁移数为 m。当只对一个迁移系统进行验证时，有 $n=n_1=n_2, m=m_1=m_2$，故在最坏情形下，局部算法验证状态对 (s,t) 是否满足模糊互模拟关系的时间复杂度为 $O(m^2n^6)$，全局算法验证状态 s、t 是否满足模糊互模拟关系的时间复杂度为 $O(m^2n^4)$。然而，当 2 个状态不满足模糊互模拟关系时，局部算法只需对迁移系统的部分状态和迁移进行遍历，即可得到结果，具有更高的效率。由于目前尚无较好的模糊数据集，采用随机生成模糊迁移系统，并随机选取迁移系统中的状态的方法，对局部算法和全局算法进行对比。用 True 表示检测的状态满足模糊互模拟关系，False 表示检测的状态不满足模糊互模拟关系。

采用 Java 语言对算法 1 进行了实现。实验环境为：Windows 10 家庭中文版，Intel® Core™ i7-7700HQ CPU 2.80 GHz，8 GiB RAM。

首先在具有相同状态数不同迁移数的模糊迁移系统下，对算法的运行时间进行比较，结果如表 1 所示。从表 1 可看出，全局算法在状态数相同迁移数相差不大的情况下，运行时间相差无几；在状态满足互模拟关系与不满足互模拟关系的情况下，运行时间也相差不大，这与全局算法的特点相同；局部算法的运行时间与其所选择判断模糊互模拟关系的状态对有关；在对模糊迁移系统中任意的 2 个状态进行互模拟等价关系判断时，局部算法比全局算法的效率更高。

表 1 相同状态数，不同迁移数算法的运行时间 s

n	m	Result	局部算法	全局算法
20	37	True	0.128 3	0.090 1
20	31	False	0.875 2	0.104 3
40	83	True	0.199 3	0.168 7
40	119	False	0.064 2	0.188 1
80	304	True	2.156 2	1.269 6
80	322	False	0.954 1	1.237 2
100	407	True	2.219 8	2.498 6
100	386	False	0.659 5	2.096 4
150	568	True	4.407 8	8.856 2
150	589	False	6.697 8	9.906 6
200	855	True	14.956 5	25.518 2
200	721	False	7.741 2	22.358 5

当对 2 个不同的模糊迁移系统是否满足模糊互模拟关系进行比较时，全局算法的时间复杂为 $O((m_1+m_2)^2(n_1+n_2)^4)$，时间复杂度高于局部算法。在对 2 个不同的迁移系统进行检测时，算法运行时间如表 2 所示。

表 2 2 个不同迁移系统算法运行时间 s

n_1	m_1	n_2	m_2	Result	局部算法	全局算法
10	22	10	32	False	0.068 3	0.124 1
30	84	30	84	True	0.151 1	0.580 9
50	158	50	158	True	0.275 2	1.753 3
70	218	70	218	True	0.405 8	4.138 2
90	243	90	252	False	0.167 2	11.238 1
100	320	100	320	True	0.748 0	19.812 5

从表 2 可看出，无论 Result 的结果为 True 还是 False，局部算法的运行效率都比全局算法的运行效率高，这与理论推导一致。

由表 1、2 可知，虽然全局算法的时间复杂度比局部算法更低，但在对一个模糊迁移系统中的一对给定状态是否满足模糊互模拟关系进行判断，或者对 2 个不同模糊迁移系统中的给定状态对是否满足模糊互模拟关系时，局部算法往往比全局算法拥有更高的效率。

5 结束语

为了可以更加高效地对指定状态是否满足模糊互模拟关系进行判定，给出了模糊互模拟等价验证的局部算法。在局部算法中，对迁移系统进行深度优先搜索遍历，并在遍历的同时对其后继状态是否满足模糊互模拟关系进行判断。该模糊互模拟局部验证算法的时间复杂度为 $O(m_1m_2n_1^3n_2^3)$，空间复杂度为

$O(n_1 n_2)$。

弱互模拟将一个给定的过程转变与另一个过程转变相匹配，同样是解决空间状态爆炸的有效途径。作为未来的工作，考虑将弱互模拟的概念引入到模糊迁移系统中，给出其定义，并对模糊弱互模拟等价验证的全局算法、局部算法和准局部算法进行研究。

参考文献：

[1] BAIER C, KATOEN J P. Principles of Model Checking [M]. Cambridge: The MIT Press, 2008: 449-582.

[2] PAIGE R, TARJAN R E. Three partition refinement algorithms[J]. SIAM Journal on Computing, 1987, 16 (6): 973-989.

[3] FERNANDEZ J C, MOUNIER L. Verifying bisimulations "on the fly"[C]//Proceedings of 3rd International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols. New York, NY: ACM, 1990: 95-110.

[4] DU W, DENG Y. A quasi-local algorithm for checking bisimilarity[C]//IEEE International Conference on Computer Science and Automation Engineering. Piscataway, NJ: IEEE Press, 2011(2): 1-5.

[5] LIN H M. "On-the-Fly Instantiation" of Value-Passing Processes[M]//Formal Description Techniques and Protocol Specification, Testing and Verification. Boston: Springer, 1998: 215-230.

[6] DENG Y, DU W. A local algorithm for checking probabilistic bisimilarity[C]//Fourth International Conference on Frontier of Computer Science and Technology. Piscataway, NJ: IEEE Press, 2009: 401-407.

[7] 张小红，裴道武，代建华. 模糊数学与 Rough 集理论[M]. 北京：清华大学出版社，2013: 1-12.

[8] DOOSTFATEMEH M, KREMER S C. New directions in fuzzy automata[J]. International Journal of Approximate Reasoning, 2005, 38(2): 175-214.

[9] CAO Y, EZAWA Y. Nondeterministic fuzzy automata [J]. Information Sciences, 2012, 191: 86-97.

[10] LI Y. Fuzzy turing machines: variants and universality [J]. IEEE Transactions on Fuzzy Systems, 2008, 16 (6): 1491-1502.

[11] BUGARIN A J, BARRO S. Fuzzy reasoning supported by Petri nets[J]. IEEE Transactions on Fuzzy Systems, 1994, 2(2): 135-150.

[12] PEDRYCZ W, GOMIDE F. A generalized fuzzy petri net model[J]. IEEE Transactions on Fuzzy Systems, 1994, 2(4): 295-301.

[13] WU H, CHEN Y. Coalgebras for fuzzy transition systems[J]. Electronic Notes in Theoretical Computer Science, 2014, 301: 91-101.

[14] WU H, DENG Y. Distribution-based behavioral distance for nondeterministic fuzzy transition systems[J]. IEEE Transactions on Fuzzy Systems, 2017, 26(2): 416-429.

[15] PAN H, ZHANG M, CHEN Y. Bisimilarity for fuzzy doubly labeled transition systems[J]. Quantitative Logic and Soft Computing, 2012: 207-214.

[16] WU H, CHEN T, HAN T, et al. Bisimulations for fuzzy transition systems revisited[J]. International Journal of Approximate Reasoning, 2018, 99: 1-11.

[17] PETKOVIĆ T. Congruences and homomorphisms of fuzzy automata[J]. Fuzzy Sets and Systems, 2006, 157 (3): 444-458.

[18] BUCHHOLZ P. Bisimulation relations for weighted automata[J]. Theoretical Computer Science, 2008, 393 (1-3): 109-123.

[19] SUN D D, LI Y M, YANG W W. Bisimulation relations for fuzzy finite automata[J]. Fuzzy Systems and Mathematics, 2009, 23: 92-100.

[20] CAO Y, CHEN G, KERRE E E. Bisimulations for fuzzy-transition systems[J]. IEEE Transactions on Fuzzy Systems, 2011, 19(3): 540-552.

[21] CAO Y, SUN S X, WANG H, et al. A behavioral distance for fuzzy-transition systems[J]. IEEE Transactions on Fuzzy Systems, 2012, 21(4): 735-747.

[22] WU H Y, CHEN Y X, BU T M, et al. Algorithmic and logical characterizations of bisimulations for non-deterministic fuzzy transition systems[J]. Fuzzy Sets and Systems, 2018, 333: 106-123.

[23] ĆIRIĆ M, IGNJATOVIĆ J, JANČIĆ I, et al. Computation of the greatest simulations and bisimulations between fuzzy automata[J]. Fuzzy Sets and Systems, 2012, 208: 22-42.

[24] IGNJATOVIC J, CIRIC M, STANKOVIC I. Bisimulations in fuzzy social network analysis[C]//Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology. Gijón: Atlantis Press, 2015: 404-411.

[25] 曾宇清，钱俊彦. 基于自动机的并发程序符号可达分析[J]. 桂林电子科技大学学报，2013, 33(4): 300-304.

[26] 叶玲玲，钱俊彦，查显伟. 基于 Büchi 自动机化简的 JavaMOP 监控器构造方法[J]. 桂林电子科技大学学报，2019, 39(5): 374-378.

编辑：张所滨